

**NETWORK COMMUNICATIONS HAVING ENDPOINT
DEVICE WITH AUTOMATIC CONNECTION TO IPBX**

Related Patent Documents

This patent document is related to U.S. Patent Application Serial No. 60/154,093 (8X8S.243P1) and entitled "Voice-Over IP Audio Terminal Processor," filed September 15, 1999, and fully incorporated by reference; and is also related to the following patent documents: U.S. Patent Application Serial No. 60/212,220 (8X8S.246PA), entitled "Communications System Architecture," U.S. Patent Application Serial No. 60/212,221 (8X8S.248PA), entitled "IP Phone Circuit Arrangement and Method," U.S. Patent Application Serial No. 09/597,705 (8X8S.249PA), entitled "Communications Controller and Method Therefor," U.S. Patent Application Serial No. 60/211,993 (8X8S.254PA), entitled "High Availability IP Telephony," U.S. Patent Application Serial No. 60/212,215 (8X8S.255PA), entitled "System Interface Implementation for Hosted iPBX," U.S. Patent Application Serial No. 60/211,992 (8X8S.256PA), entitled "IP Telephony Station Equipment," and U.S. Patent Application Serial No. 60/212,219 (8X8S.257PA), entitled "iPBX Hosting," each of which was filed with the U.S. Patent and Trademark Office on June 16, 2000 and is herein fully incorporated by reference.

Field of the Invention

The present invention relates to communication systems, and more particularly, to an internet-type telephony endpoint device and communications methodology therefor.

Background of the Invention

The electronics industry continues to rely upon advances in technology to realize higher-functioning devices at cost-effective prices. For many communication applications, realizing higher-functioning devices in a cost-effective manner requires the creative use of communications channels. Many technologies have been developed that have enhanced communications. Examples include the Internet, facsimile applications, public switched telephone networks (PSTN), wireless telephones, voicemail systems, email systems, paging systems, conferencing systems, electronic calendars and appointment books, electronic address books, and video-image processing systems that communicate video data simultaneously with voice data over a telephones and the Internet. As the popularity of these technologies increases, so does the need to merge and coordinate these technologies in a manner that is convenient and cost-effective for the user.

The above-mentioned technologies have been developed in a relatively isolated manner. Large-scale integration of multiple communications systems has been costly and difficult to achieve and manage. One difficulty stems from the variety of communications channels and data types used for various applications. For example, telephony signals can now be transmitted by methods and systems including traditional publicly-switched telephone networks (PSTN), Internet telephony service providers (ITSP), packet-based systems, digital wireless systems, analog wireless systems, private branch exchanges (PBX), cable systems, T1 systems, integrated service digital network (ISDN), and digital subscriber line (DSL) systems, to name a few. Many telephone systems, particularly for business applications, offer services including voicemail,

facsimile, call forwarding, and other call-controls, but these systems are often costly, difficult to manage, limited in scope, and do not offer integration of various communications methods. In addition to difficulties inherent in coordinating telephony-type communications, the coordination of additional communications, such as text, video, or other data, provides additional challenges.

Widespread acceptance and usage of communication systems and services are largely a function of cost and user convenience. Therefore, widespread acceptance and usage of such technology cannot be forced, even when appropriately addressing the marketing needs and overcoming the exorbitant costs of the mass production equipment.

The scalability of a communications system weighs heavily upon the acceptance of the system. As the face of today's workplace is changing, the ability to provide flexible communications services is becoming increasingly important. Many employees are highly mobile, moving between companies and between jobs within a company. When employees are added, leave or move within the company, the communications systems for those employees must be modified. In addition, many employees work from several locations, such as a base office, home, or a branch office. To accommodate ongoing communications needs, a user-friendly and user-reconfigurable system would be advantageous.

Summary of the Invention

The present invention is directed to a telephony communications arrangement and device implementing an autodiscover feature that enables an endpoint device to automatically establish connection with a desirable server in a web of interconnected

servers. In addition, the ease of use and cost-effectiveness of the present invention enable the use of such communications control and coordination many applications, including small and medium-sized business applications. The present invention is exemplified in a number of implementations and applications, some of which are summarized below.

According to an example embodiment of the present invention, a telephony communications arrangement includes an internet-based private branch exchange with a programmable processor circuit programmed to control a server at the internet-based private branch exchange. The server is adapted to communicate to a remote location over a first communications path using a plurality of packet-based communications and endpoint devices. The packet-based communicating endpoint devices are adapted to communicate with the internet-based private branch exchange over a second communications path which is directly coupled communicatively to the first communications path. The second communications path is also communicatively coupled to the plurality of other packet-based servers. Each packet-communicating endpoint device is configured and arranged to automatically broadcast its identity and establish communication with the internet-based private branch exchange from the plurality of other packet-based servers for establishing packet-based communications between the packet-communicating endpoint device and the internet-based private branch exchange.

According to other aspects of the present invention, one or more of the packet-communicating endpoint devices are adapted to seek one of the internet-based private branch exchanges for establishing a communication link. This seeking mode is implemented in different manners, according to different implementations of the present

invention. For example, one implementation involves the endpoint device being adapted to broadcast its identity in anticipation of a DNS (Directive Name Server) being previously configured to monitor for such broadcasts. In response to detecting this broadcast, the DNS responds with an assignment of the iPBX for the broadcasting endpoint device. The broadcasting endpoint device then commences communication with the assigned iPBX.

In another example implementation, the endpoint device broadcasts its identity and a unique iPBX is configured to monitor for such broadcasts. In response to detecting this broadcast, the iPBX responds by informing the broadcasting endpoint device that all subsequent communication should be directed to this unique iPBX.

Respective variations of the above two specific example implementations include further programming the endpoint device with a security code that is used by the monitoring device to reduce the likelihood that the endpoint device would be improperly assigned by either the broadcast-monitoring DNS or the broadcast-monitoring iPBX.

In another example implementation, the packet-communicating endpoint devices are implemented using a forced "address-search" for identifying the internet-based private branch exchange relative to the plurality of other packet-based servers. In this environment, each of the packet-communicating endpoint devices is pre-programmed with the identification of the DNS for the purpose of submitting to the DNS a request for the appropriate iPBX assignment.

According to another example embodiment of the present invention, each of the packet-communicating endpoint devices is further adapted to store a unique Media Access Call address and is able to communicate the unique Media Access Call address

with the internet-based private branch exchange. Further, each of the packet-communicating endpoint devices may be adapted to store a unique code that identifies the internet-based private branch exchange relative to the plurality of other packet-based servers.

In another example embodiment, each of the packet-communicating endpoint devices is further adapted to execute a program that causes the packet-communicating endpoint device to establish communication based on a set of search rules. For example, the endpoint device for one of the servers that manifests an acceptable routing path to establish packet-based communication. This embodiment may be further modified to define the acceptable routing path in terms of an optimally minimum number of routing connections identified over a predetermined time period during a broadcast effort for establishing connection with the desired iPBX.

The above summary of the present invention is not intended to describe each illustrated embodiment or every implementation of the present invention. The figures and detailed description that follow more particularly exemplify these embodiments.

Brief Description of the Drawings

The invention may be more completely understood in consideration of the following detailed description of various embodiments of the invention in connection with the accompanying drawings, in which:

FIG. 1 is an illustration of a packet-based communications system, according to an example embodiment of the present invention;

FIG. 2 is another illustration of a communications system, according to another example embodiment of the present invention;

FIG. 3 is an illustration of an endpoint communications device, according to another example embodiment of the present invention, communicating with an internet-type private branch exchange; and

FIG. 4 is an illustration of another endpoint communications device, according to another example embodiment of the present invention, communicating with an internet-type private branch exchange.

While the invention is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the invention to the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

Detailed Description

The present invention is believed to be applicable to various types of communications systems, and has been found particularly suited to such systems requiring or benefiting from user-friendly control for processing various telephony communications data types and routing the communications via an Internet protocol (IP) type of network. For example, the present invention has been found to be advantageous when operating in conjunction with an iPBX server of the type characterized in the above-referenced patent documents. While the present invention is not necessarily

limited to such systems, various aspects of the invention may be appreciated through a discussion of various examples using this context.

According to an example embodiment of the present invention, a communications service provider network is adapted to receive, process and deliver communications data of various types using a cost-effective, user-friendly operations platform. The network includes a plurality of communication stations communicatively coupled to one or more communications links. One of the communication stations is adapted to communicate packet-based data to a computer server arrangement communicatively coupled to the communications links. The telephony-controlling computer server can be configured in any of a number of ways including, for example, being adapted to operate with a IP telephony switch by passing and/or redirecting data-carrying messages for uniquely-targeted endpoint devices (*e.g.*, an IP telephone, a display or household appliance). The server and/or IP telephony switch is communicatively coupled to the communications stations (and in certain applications other endpoint devices) through a communications links which couples to a multitude of other server-type communication terminals. One (or multiple ones) of the communication stations is programmed to automatically discover a selected one of many possible servers for establishing a link therewith. This automatic discovery is achieved, for example, by designing (*e.g.*, programming) the endpoint device to respond to activation by automatically broadcasting its identity and programming a DNS or a targeted iPBX to monitor the network for such a broadcast and then assign the broadcasting endpoint device with the identity and any desired related codes for establishing communication between the endpoint device and the targeted

0056793-0100

iPBX. Such an assignment from the DNS can be direct or through another server such as the targeted iPBX.

Also according to the present invention, the above-described autodiscover mode is implemented to permit ease of operation for the user of the IP phone. In a particular example embodiment, for example, the autodiscover protocol is implemented by configuring the server-seeking endpoint device with a unique code that identifies the internet-based private branch exchange relative to the plurality of other packet-based servers. In another example embodiment, the autodiscover mode is implemented by forcing a search for a server having one or more selected characteristics based on a set of rules with which the server-seeking endpoint device is programmed to abide by. In another example embodiment, this mode is achieved using a combination of selectable modes, including for example using one of the above approaches as a default mode and each other mode being selectable by the IP phone user or another personnel responsible for the programming and/or maintaining operation of the IP phone.

FIG. 1 is a specific example communications network operating consistent with the above discussion and according to the present invention, and is useful in obtaining a better appreciation of the present invention. In this system, a telephony communications arrangement includes a number of service providers 20, 22, 24 and 26, with the service provider 26 being adapted as a target internet-based private branch exchange providing communications operations with a target set of endpoint devices and another of the service provides being a DNS for the local Web network. The service provider 26 includes a programmable processor circuit (not shown in FIG. 1) programmed to control the server and internet-based private branch exchange operations; these operations

include communicating to remote locations over a first communications path, such as a conventional telephony path (*e.g.*, ISDN, cable, wireless, POTS), linking the service provider 26 as one of a plurality of possible communication targets for IP interface 30. The WEB, depicted as 32 of FIG. 1, is an example of such a first communications path; other examples include those paths that provide more private or secure communications, such as intranet LAN-type applications and government-defined communication networks.

The IP interface 30 is depicted in this example illustration as providing a communications path to the WEB 32 for various endpoint devices such as internet-protocol (IP) phones 41-44, which are interconnected using a second communications path, such as a LAN or other type, provided within one or more facilities hosting the IP phones 41-44. Any of the internet-protocol phones 41-44 can be implemented with an auto-discover mode for seeking a particular type of server, such as a target iPBX-based server 26. In this context, the target set of endpoint devices for the target iPBX-based server 26 includes at least IP phone 41. In one example specific application embodiment, the interface 30 of FIG. 1 is a circuit implementation available from Netergy Networks, Inc., ^{such} as characterized in connection with the ^{first (and other of the)} above-referenced patent documents.

In this auto-discover mode, the IP phone automatically searches for and distinguishes the internet-based private branch exchange 26 from the plurality of other packet-based servers 20, 22 and 24 and, in response, establishes packet-based communications with the targeted server.

Implementing the internet-protocol phone 41 with an auto-discover mode is useful in many regards. For example, should a facility location be assigned with an IP phone 41

for the first time, a communications manager (*e.g.*, MIS personnel or an employee responsible for programming/maintaining the IP phone 41) can configure the IP phone 41 so that each time the IP phone attempts to communicate outside the facility or facilities, it searches for a particular server that is configured to remotely manage various aspects of the IP phone 41, for example, as described in connection with the above-referenced patent documents. In this manner, the user of the IP phone 41 does not have to possess any previous knowledge of the target server's IP address or under which circumstances the user should be connecting to one server versus another server.

In another embodiment, the interface 30 of FIG. 1 is programmed and configured to perform the search tasks, including broadcasting functions for example, for implementing this autodiscover mode on behalf of the IP phone 41.

FIG. 2 is another specific example communications network, also according to the present invention, which illustrates another application and other aspects of the present invention. In this communications environment, a service provider 110 is adapted to control and route communications data for a user premise 120. User premise 120 includes a variety of communications devices including an IP phone 122, computers 124 and 126, and two telephones 132 and 134 coupled to an IP gateway 130. Also shown is an IP appliance 135 adapted with a configuration 136 to seek a particular type of service provider, such as the service provider 110 shown in FIG. 2. Each of the devices is communicatively coupled to the service provider via a router 140 and a communications link 150. The service provider is adapted to send and receive communications data via a router 160 coupled to the communications link 150. The router 160 is coupled to the

Internet, to a server 170 and an IP/PSTN gateway 180. The router 160 is adapted to route communications to either the router 140, the Internet, the server or the gateway 180.

The server 170 is programmed to control the routing of communications data to and from the user premise 120. The programming is accomplished in various manners, depending upon the functions being programmed and the security access level of the programming source. First, individual users at the customer premise can program the server. Each user programs various communications selections, such as those described hereinabove. Inputs from the computers 124 and 126, from the IP telephone 122, or from the telephones 132 and 134 are all used for controlling the server. The computers have a user interface adapted to provide various communications selections to the server. The user interface may include, for example, the graphical user interface described in U.S. Patent Application Serial No. 09/597,704 (8X8S.249PA), filed on June 16, 2000, and entitled "Communications Controller and Method Therefor." Selections are made at the computer interface for controlling communications between the computer and one or more other communications devices. Similarly, the IP telephone 122 and other telephones 132 and 134 are used to input control information, such as via a touch-tone sequence or other control code entry.

In addition to programming the server at the user premise 120, the server may also be programmed at remote locations, such as at a communications device communicatively coupled to the Internet or to the PSTN. As discussed in connection with communications devices located the user premise, various control inputs are provided to the server via the respective connections using remote communications devices. For example, Internet communications devices such as a computer, a wireless

telephone having Internet communications ability or an Internet interface such as a WebTV interface could all be adapted for use in communicating with the server to provide programming information.

For further details regarding manners of implementing the system of FIG. 2, reference may be made to U.S. Patent Application Serial No. 60/212,159 (8X8S.247PA), filed on June 16, 2000, and entitled "Communications Service Provider Network."

FIGs. 3 and 4 respectively show first and second example implementations for the endpoint device 41 of FIG. 1, or for endpoint appliance 135 of FIG. 2. With reference to each of these devices 41 and 135, each of FIGs. 3 and 4 includes the corresponding notation 41' and, for the related iPBX, 26'. It will be appreciated that certain of the elements illustrated in FIGs. 3 and 4 are not required for all possible endpoint or appliance types. For example, an endpoint device that does not receive and process user audio does not require an audio input port and related audio-input processing circuitry.

The endpoint device 41' includes audio I/O ports (not shown) and related processing circuitry 202, implementations of which are discussed for example in connection with U.S. Patent Applications, Serial No. 09/086,434 (8X8S.200PA), filed on May, 28, 1998, and Serial No. 09/392,124 (8X8S.239PA), filed on September 8, 1999, and Serial No. 09/203,311 (8X8S.217PA), filed on December 1, 1998. Various implementations for the user-output and user-input implementations 204 and 206 are well known and can be implemented using various technology including the conventional technology presently available on conventional telephones. The endpoint device 41' also includes a voice-over-IP interface circuit 210, which is used to present internet protocol data from the iPBX 26' to the display 204 and/or to the processing circuitry 202. In the

implementation of FIG. 3, the voice-over-IP interface circuit 210 is also used to present user-input data from the user-input implementations 204 and 206 of the device 41' to the iPBX 26'.

A significant difference between the implementations of FIGs. 3 and 4 is that the implementation of FIG. 4 includes a circuit configured for special executive-decision processing; this block is denoted 220. The tasks executed by the circuit 220 depend upon the particular implementation and in certain applications these include analyzing and validating security in connection with iPBX assignments from over the Web. The skilled artisan will appreciate that, for the less-complex endpoint devices, the implementation of FIG. 3 is less expensive to design and manufacture.

Using the above or another configuration, the packet-communicating endpoint devices seek a targeted one of the internet-based private branch exchanges for establishing a communication link based a previously-assigned unique server code assignment, for example, as currently assigned by the Internet naming authority. Using this unique server code, the iPBX-seeking mode is implemented in different manners. For example, one specific implementation involves the endpoint device being adapted to broadcast its identity (e.g., using its Internet-assigned MAC address) in anticipation of a DNS (Directive Name Server) being previously configured to monitor for such broadcasts. In response to detecting this broadcast, the DNS responds with an assignment of the iPBX for the broadcasting endpoint device. The broadcasting endpoint device then commences communication with the assigned iPBX.

In another specific implementation, the endpoint device broadcasts its identity and a unique iPBX is configured to monitor for such broadcasts. In response to detecting this

broadcast, the iPBX responds by informing the broadcasting endpoint device that all subsequent communication should be directed to this unique iPBX.

For each of these implementations, either of the endpoint-device implementations shown in FIG. 3 or FIG. 4 can be used. For further details of these implementations, reference may be made to the attached entitled, **Symphony MGCP DNS Call Agent Discovery** (Appendix A), and to **VoIP Terminal Discovery Protocol**, (Appendix B); each being fully incorporated herein by reference.

In yet another specific example implementations, the packet-communicating endpoint devices seek a targeted one of the internet-based private branch exchanges for establishing a communication link based on the previously-assigned unique server code assignment being programmed into the respective endpoint devices, for example, as programmed by the facility MIS manager before installation in the network. At such a time (and also with the previous implementations), the facility MIS manager can also program a security or password code for validation to a responding serving node on the network that might attempt to reassign the server code or otherwise establish communication with the endpoint device. After the endpoint device broadcasts its identity, a monitoring DNS or the target iPBX can respond with specific instructions for communication. The broadcasting endpoint device then commences communication based on the assignment(s) being provided.

While the present invention has been described with reference to several particular example embodiments, those skilled in the art will recognize that many changes may be made to the present invention. For example, other embodiments of the present invention can include a combination of one or more aspects discussed herein or as

[illegible]

App K. A

Symphony MGCP DNS Call Agent Discovery

Introduction

This document describes how a Symphony MGCP MTA can automatically discover its Call Agent address using DNS. The process is designed to be the same as the H323 Gatekeeper DNS discovery procedure for commonality of provisioning between the two protocols. The bulk of this document is a MGCP specific version of the H323 documentation.

The SRV resource record query

The first solution uses the fact that the call agent is basically a system service, and the transport address of a named system service can be extracted from DNS by using a query for a new type of DNS Resource Record, called SRV (for "service location record"). Given a domain name, an SRV record query will be made for the transport address of the mgcp service for that domain. The format of the response is the same as for a TXT record query documented below.

This simple solution will soon be standard. The problem is that almost no current DNS client or server implementations support the SRV resource record yet. Unless the DNS client knows about the SRV resource record type, it is not possible for it to pass on queries for this resource record. Until this support becomes widespread, there is a reasonable chance that the SRV query will fail.

The TXT record query

All current DNS implementations support the TXT resource record. Basically this is some free text that can be returned for each domain name. It is possible to store many TXT resources for a single domain. The standard stipulates that all TXT records will be returned when a query is made for them. The domain name is used to make a DNS TXT query for that domain. The returned resource records are lines of free text, and the terminal will then look for lines in the response of the form:

mgcp [<ca>@]<domain name>[:<portno>] [<priority>]

The <ca> field is an optional call agent ID which is separate from the domain name. If this field is missing, then it is assumed to be ca.

The <domain name> field can be either the name of the A-record which contains the call agents IP address, or a raw IP address in dotted form. The domain name need not be fully qualified; if it is not, the subdomain in which the TXT record was found should be appended to it to form the fully qualified A-record name.

The optional [:<portno>] can be used to specify a port number other than the standard mgcp port.

The optional [<priority>] field specifies the order in which the listed call agents should be accessed for discovery queries if there is more than one mgcp TXT record. Lower numbers have higher priority.

White spaces are used as delimiters between mgcp and ca if present or domain name, and between portno and priority. White spaces consist of any number of spaces or tabs.

Examples of valid gatekeeper TXT records:

mgcp agent

mgcp ca@agent.company.com

mgcp agent:1500 3

mgcp 172.11.22.33:1500 2

Since DNS requires a server to return all TXT records associated with a domain name, the client can filter out and process only those records which are useful to it. It also allows DNS to return an ordered list of call agents which can serve as alternatives and back-ups.

Note that the server returned in such a query might be an actual transport address in dotted decimal notation, or it could be an FQDN which itself requires an A-record query in DNS to determine the transport address. The advantage of using an FQDN is the usual hiding of actual IP numbers. The advantage of using IP numbers is that a second DNS query is avoided, thus speeding up this pre-call setup time.

[illegible]

APPX. B

VoIP Terminal Discovery Protocol

Introduction

In deploying and provisioning a Voice over IP (VoIP) solution it is necessary to both configure the central PBX and the terminals. While each terminal can be individually provisioned it is often easier for an administrator to deploy unconfigured units and remotely provision them at a later time. With Symphony VoIP terminals this provisioning process can be achieved by remotely accessing the built in web based configuration. However before the built in web configuration can be accessed the IP address of the Symphony units must be determined. This document describes a mechanism for automatic network discovery of VoIP terminals and their IP addresses.

Obtaining a network address

Before a terminal on the network can be accessed it needs to acquire a unique network address. DHCP (Dynamic Host Configuration Protocol) described in RFC1531 provides an industry standard method for dynamic allocation of IP addresses. Within DHCP addresses can be dynamically allocated or statically provisioned by the network administrator at the DHCP management console depending on the desired network topology. This allows address mappings to be easily redesigned from a central system. A single DHCP server can also serve more than one subnet by utilizing the BOOTP relay service provided by network routers.

Symphony VoIP terminals support automatic network address acquisition using DHCP. This gives plug and play benefits allowing an untrained user to simply plug-in the unit and have it network aware. Unfortunately on a device that does not have a display it is difficult to determine the IP address that the unit has acquired without administrative access to the DHCP server lease log. This document describes how this limitation can be removed.

Automatic discovery

Automatic discovery can be achieved by use of an IP multicast group. On startup each VoIP terminal joins the VoIP-discovery multicast group and waits for a discovery message. When a user at an administrative console wishes to find the active VoIP terminals the console transmits a discovery message to the multicast group. Each terminal then responds with its information (IP address and MAC address) in a unicast packet back to the admin terminal.

In order to prevent a flood of messages the admin terminal can specify the following options in the discovery message:

- Time to respond. Each terminal will randomly choose a time between 0 and the specified time to respond before sending its response. This spreads out the messages and reduces the peak bandwidth required.
- Respond only if "on time" less than X. Each terminal will only respond if it has been powered up less than the specified "On time". This could be used by an admin terminal to hunt for recently installed units only (maybe poll every hour for units powered up within the hour).
- Time to live. This limits how many 'hops' the message will make through routers before being discarded. In normal operation a TTL of 1 will normally be used (ie. Restrict to local segment).

Use of a multicast group provides the most efficient way of reaching many sparsely distributed nodes. Unlike ethernet broadcasts which are received by all nodes on a segment, multicast is only received by nodes that are listening for the specific multicast group selected. In addition multicast messages are capable of being passed through routers so that the discovery can scan beyond the local subnet.

Discovery packet

This packet is sent (UDP) to the multicast group to initiate discovery process. All values are sent in network order (big endian). A receiving node must be capable of ignoring any future extensions (ExtensionsPresent != 0) and correctly parsing the rest of the packet. A sending node will set an appropriate TTL for the multicast packet (TTL setting is network API dependant).

Offset	Size	Field Name	Description
0	WORD	Opcode	= 0x41 VOIP_DISCOVERY
2	WORD	Length	Length of entire packet
4	WORD	Checksum	Checksum of entire packet
6	OCTET	ExtensionsPresent	= 0x00 NO_EXTENSIONS_PRESENT
7	OCTET	ResponseTime	Time to respond. Each terminal will randomly choose a time between 0 and ResponseTime (in seconds) to respond before sending its response.
8	WORD	OnTime	Respond only if powered up less than OnTime (in seconds). A value of 0 means all respond.
10	WORD	ReportToPort	Port number to send response to
12	DWORD	ReportToIPAddress	IP address of node to send response to
16		...	(future extensions)

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the packet. For purposes of computing the checksum, the value of the checksum field is zero.

Discovery Response packet

This packet is sent (UDP) unicast to the ReportTo address/port specified in the discovery process. A receiving node must be capable of ignoring any future extensions (ExtensionsPresent != 0) and correctly parsing the rest of the packet.

Offset	Size	Field Name	Description
0	WORD	Opcode	= 0x42 VOIP_DISCOVERY_RESPONSE
2	WORD	Length	Length of entire packet
4	WORD	Checksum	Checksum of entire packet
6	OCTET	ExtensionsPresent	= 0x00 NO_EXTENSIONS_PRESENT
7	OCTET	Reserved	(alignment/reserved for future use)
8	DWORD	ProtocolsSupported	Protocols supported (see table below)
12	DWORD	IPAddress	IP address of the responding node (will normally match the source IP address of the packet)

16	6 OCTETS	MacAddress	MAC address of the responding node
22	WORD	Reserved	(alignment/reserved for future use)
24		...	(future extensions)

ProtocolsSupported is a bit field with any of the following bits set. All other bit values are reserved for future allocation.

Protocol	Mask	Description
MGCP	0x00000001	MGCP supported
H323	0x00000002	H323 supported
HTTP	0x00000004	HTTP (web server) supported
SNMP	0x00000008	SNMP supported
TFTP	0x00000010	TFTP supported
(extension)	0x80000000	Reserved for indicating future extensions

System requirements

The systems must meet the following requirements to be able to implement the mechanism described above:

- Administrative terminal must be able to transmit to a multicast group.
- VoIP terminal must be able to join a multicast group. This implies:
 - The ability to receive and filter multiple multicast mac addresses
 - Support of IGMP protocol to notify routers of multicast group membership
 -
- Network must support DHCP allocation of IP addresses

Network numbers

Multicast group address	224.0.1.149 (assigned by IANA)
Multicast group port	4444

DHCP Server

Accessing configuration information

Setting configuration parameters

```
String msg = new String("POST xxx.htm \r\ncontent-length:15\r\n\r\nSYSID=216ALPHA&\r\n");
```

```
InetAddress dest = InetAddress.getByName("207.82.37.216");
```

```
Socket s = new Socket(dest,80); // connect to webserver
```

```
OutputStream os = s.getOutputStream();
```

```
os.write(msg.getBytes());
```

Reading configuration parameters

```
String msg = new String(
```

```
"POST xxx.htm \r\ncontent-length:27\r\n\r\n\r\nREDIRECT=$$SYSID$$&\r\n";

InetAddress dest = InetAddress.getByName("207.82.37.216");

Socket s = new Socket(dest,80); // connect to webserver

OutputStream os = s.getOutputStream();

InputStream is = s.getInputStream();

os.write(msg.getBytes()); // send request

byte[] rcv = new byte[1000];

int len = is.read(rcv); // read response
```

Note the variable name must be surrounded by \$\$ - this marks a variable for substitution on transmission from symphony. Multiple variables can be retrieved simultaneously by adding them to the substitution list. For example REDIRECT=\$\$\$SYSID\$\$?\$\$CALLAGENTIP\$\$. For each connection made only one POST can be sent before the socket must be closed (http server processes only a single request on any connection). Standard URL character encoding (see RFC1738) should also be applied to the string that is sent (not shown in above example).

Writing a unique identifier to Symphony

The Symphony V2.0 http protocol will allow Intraswitch to write a unique identifier to the Symphony configuration store. This may provide a superior method of identifying Symphony's than using the MAC address, which will change if a unit fails and is replaced. Lack of an identifier can be used to indicate that Intraswitch has never configured this Symphony.

Multicast transmission from Java

Transmission to a multicast group from Java code is accomplished using code similar to the following. Note: setTimeToLive defaults to 1 for a multicast address – this must be set larger if you wish to multicast beyond the local subnet/router.

```
byte[] msg = {1,2,3,4,5,8,8};

InetAddress group = InetAddress.getByName("230.230.230.230");

MulticastSocket s = new MulticastSocket();

DatagramPacket pkt = new DatagramPacket(msg, msg.length, group, 4444);

s.setTimeToLive(40); // set the TTL for the multicast socket

s.send(pkt);
```